

## Potential Distribution and Multi-Terminal DC Resistance Computations for LSI Technology

*Computer time and storage requirements are the two main considerations in the design of a packaging analysis software tool for the problem of calculating the electric potential distribution in arbitrary geometrical shapes. The FEM (Finite Element Method) is the accepted approach for solving such problems. A new formulation for the linear triangular element is presented which is used to derive a very simple and computationally inexpensive linear rectangular element equation interrelating only the geometrical centers of the elements. The result is a much sparser assembly matrix with a maximum of five non-zero entries per equation compared with the usual nine of the FEM formulation. In addition, a method to obtain the minimum bandwidth of the matrix is given for the efficient and static use of external storage, permitting the solution of any size problem. The methods are applicable to multi-plane, multi-terminal configurations for the production of equivalent-resistance networks and for the calculation of the potential distribution throughout the configurations.*

### Introduction

Today, accurate electrical evaluation of any proposed circuit packaging scheme, at any level of packaging, is essential to the success of a given LSI (Large Scale Integration) technology. This is necessitated by the rather tight tolerance for the allowable voltage loss from the power supply point to the circuits on the chip. In addition, the complexity of geometries of the current-conducting planes, the existence of numerous contact points or terminals representing different boundary conditions, and nonuniformity in the material properties make an analytic solution all but impossible. The simplifying assumptions that have to be made quite often result in a model which bears little resemblance to reality. Aside from the integral equation method, the principal numerical techniques for solving complex boundary problems are the finite difference and the finite element methods [1-4]. A new computational technique, based on a numerical approach using the principles of current conservation and continuity, has been developed for calculating the potential distribution for arbitrary geometry and producing an equivalent resistance network as a model of the geometry. Given voltage and/or current conditions, the network

can be solved, or it can be merged with other electrical data for use with a general circuit analysis program.

In the finite difference method, a grid of points, usually rectangular, is superimposed on the region to be analyzed and the given differential equation is approximated by a corresponding difference equation. This equation interrelates the values of the function at adjacent mesh-points. Application of the difference equation to each mesh-point produces a set of algebraic equations to be solved for the unknown values of the function at each point. The method is well suited for orthogonal geometries and regions of uniform electrical properties, but certain difficulties arise when the shapes involved have nonorthogonal sides and the region is made up of irregular geometries with different square resistances and different boundary conditions, such as constant voltage or current sources or sinks.

In the finite element method, the region under consideration is first divided into a finite number of known geometrical shapes, called elements, and a form is assumed

**Copyright** 1979 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

for the solution function, usually polynomial, within and on the boundary of the elements.

This approximating function, sometimes called an interpolating function, is defined in terms of specific points, called nodes, usually at the boundaries of the element. The degree of the polynomial approximation is related to the number of these points. Numerical element equations are then derived by minimizing the *associated functional* of a variational statement or, when a functional does not exist [4], through the use of a *weighted residual* or *energy balance* approach. Finally, the element equations are assembled and a system of equations is obtained representing the given region.

The finite element method is a powerful technique since, unlike the finite difference method, it can easily handle complicated geometries with variable material properties and mixed boundary conditions. Caution must be exercised, however, when elements of different shapes are used to discretize the continuum. Their interpolating functions must be compatible at the boundaries of different elements. It can also be argued that when the method is applied according to its standard procedure (forming the element equations with no regard to their relative position), it will be more expensive to form the system equations, compared to the finite difference method where the difference equation is written directly for every nodal point.

The method presented in this paper closely resembles the finite element method. It differs, however, in the derivation of the element equations, in which the continuity equation, the current conservation principle, and steady state conditions are used. Triangular and rectangular elements are used to discretize the region and linear approximations are assumed for the interpolating functions. Due to the  $x$ - $y$  orientation of the rectangular elements, a very simple numerical expression is derived which interrelates only the functional values at their geometrical centers. The effect is a much sparser matrix of the equations for the rectangular regions, with a maximum of five non-zero entries instead of nine for the FEM [2-4]. The two main considerations for efficiently solving the system equations are the minimization of the fill-in during factorization [5-8] and a provision to use external storage so that any size problem can be solved. The modified banded approach in decomposing the matrix combined with a novel optimal numbering scheme presented in the paper meets these requirements effectively. Regardless of the approach used in formulating the element equations, the method presented here, like the finite element method, follows a precise step-by-step procedure. The organization of this paper generally follows this procedure in describing the dif-

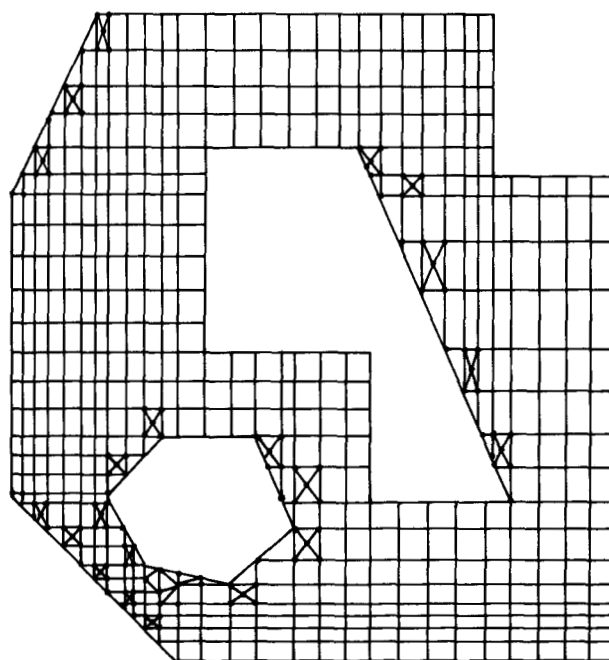


Figure 1 Discretization of the solution region into triangular and rectangular elements. Triangular elements are generated only at oblique sides of the given geometries.

ferent parts of the computer program or analysis package which has been implemented using this method.

### Discretization of the continuum

The mathematical formulation of the problem is directly related to the choice of elements and nodal points used to discretize the solution region. The number of nodal points per element is determined by the degree of the interpolating polynomial chosen to approximate the solution function within and on the boundary of the elements. In this work triangular and rectangular elements are used to discretize the region. A linear approximation is assumed for the solution function within the triangular elements, with its vertices being the nodal points. For the rectangular elements, due to the geometrical symmetry (Fig. 1) of the elements and the way the problem is formulated, a simple numerical equation has been derived which interrelates only the geometrical centers at adjacent rectangular elements. This is another place where the present method deviates from the classical finite element approach.

An automatic element generator has been developed which divides any irregular geometry with any number of superimposed shapes representing areas of different material properties, different boundary conditions, or vacant spaces, into triangular and rectangular elements matching

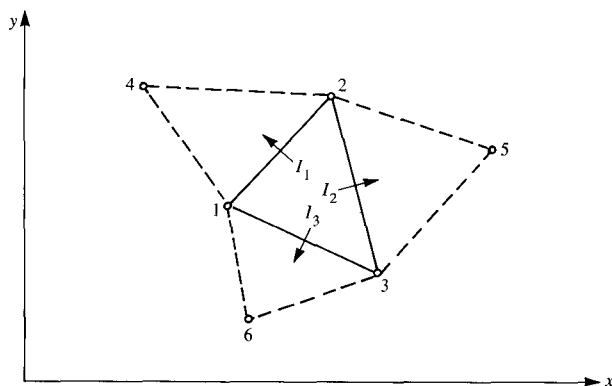


Figure 2 The element equation for the triangular center element is  $I_{\text{element}} = K^T \phi$ . It interrelates the nodal values at all six vertices.

as closely as possible the boundaries of the given geometries. As seen in Fig. 1, triangular elements are generated only at irregular boundaries of the given geometries, thus keeping their number to a minimum. A randomly oriented triangular element is computationally more expensive than an  $x$ - $y$ -positioned rectangular element. Every element is bounded on each side by only one element and no element overlaps two different shapes. A minimum amount of information is requested from the user. The shapes, what they represent, and the number of elements desired are the only input data to this module. The generator creates all the geometrical and electrical properties of the elements as well as linkage information of the nodal points required by the subsequent element formulation.

### Derivation of the finite element equations

#### • Triangular elements

A linear approximation is assumed for the potential inside and at the boundaries of the element. The vertices of the element have been chosen as the nodal points. Thus, if

$$\phi(x, y) = k_1 + k_2x + k_3y, \quad (1)$$

then from Fig. 2

$$\phi_1 = k_1 + k_2x_1 + k_3y_1,$$

$$\phi_2 = k_1 + k_2x_2 + k_3y_2,$$

$$\phi_3 = k_1 + k_2x_3 + k_3y_3. \quad (2)$$

Solving Eqs. (2) in the standard finite element formulation [4] for  $k_1$ ,  $k_2$ , and  $k_3$ , substituting into Eq. (1) and rearranging terms, we obtain

$$\phi(x, y) = L_1(x, y)\phi_1 + L_2(x, y)\phi_2 + L_3(x, y)\phi_3, \quad (3)$$

where

$$L_1(x, y) = \frac{(a_1 + b_1x + c_1y)}{2\Delta},$$

$$L_2(x, y) = \frac{(a_2 + b_2x + c_2y)}{2\Delta},$$

$$L_3(x, y) = \frac{(a_3 + b_3x + c_3y)}{2\Delta}, \quad (4)$$

$$2\Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \text{twice the area of triangle 1 2 3,}$$

and

$$a_1 = x_2y_3 - x_3y_2,$$

$$b_1 = y_2 - y_3,$$

$$c_1 = x_3 - x_2, \quad (5)$$

while the other coefficients are obtained by cyclical permutation of the subscripts. Notice that

$$\frac{\partial \phi}{\partial x} = \frac{\partial L_1}{\partial x} \phi_1 + \frac{\partial L_2}{\partial x} \phi_2 + \frac{\partial L_3}{\partial x} \phi_3,$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial L_1}{\partial y} \phi_1 + \frac{\partial L_2}{\partial y} \phi_2 + \frac{\partial L_3}{\partial y} \phi_3, \quad (6)$$

or, using Eq. (4),

$$\frac{\partial \phi}{\partial x} = \frac{(b_1\phi_1 + b_2\phi_2 + b_3\phi_3)}{2\Delta},$$

$$\frac{\partial \phi}{\partial y} = \frac{(c_1\phi_1 + c_2\phi_2 + c_3\phi_3)}{2\Delta}. \quad (7)$$

The two relations (7) will be used later in the derivation of the triangular element equation, which will be different from the derivation of the standard finite element approach.

Next, an equation will be derived that expresses the total current crossing a randomly oriented line within the solution region.

Consider side AB in Fig. 3 separating two triangular elements with material properties  $\rho_1$  and  $\rho_2$  (square resistances) and interpolating functions  $\phi_1(x, y)$  and  $\phi_2(x, y)$ . Due to continuity of current across the boundary of the two different material regions, the interface condition must hold:

$$\frac{1}{\rho_1} \frac{\partial \phi_1}{\partial \mathbf{n}} = \frac{1}{\rho_2} \frac{\partial \phi_2}{\partial \mathbf{n}}, \quad (8)$$

or, since  $\rho_1$  and  $\rho_2$  are constant inside each element,

$$\frac{1}{\rho_1} \int_A^B \frac{\partial \phi_1}{\partial \mathbf{n}} dl = - \frac{1}{\rho_2} \int_B^A \frac{\partial \phi_2}{\partial \mathbf{n}} dl, \quad (9)$$

where  $\mathbf{n}$  is the normal vector to the line AB and  $dl$  is along the interface AB. Equation (9) relates two different expressions for the total current crossing the line AB, which may be written as

$$I_{AB} = -I_{BA}. \quad (10)$$

Thus, if  $I_{12}$  is the total current crossing side AB from element 1 to element 2, then

$$I_{12} = I_{AB} = \frac{1}{\rho_1} \int_A^B \frac{\partial \phi_1}{\partial \mathbf{n}} dl \quad (11)$$

and

$$I_{12} = -I_{AB} = -\frac{1}{\rho_2} \int_B^A \frac{\partial \phi_2}{\partial \mathbf{n}} dl. \quad (12)$$

Next the integral

$$\int_A^B \frac{\partial \phi_1}{\partial \mathbf{n}} dl$$

will be evaluated.

The directional derivative of  $\phi_1$  in the direction of the normal vector  $\mathbf{n}$  is given by

$$\nabla_{\mathbf{n}} \phi_1 = \frac{\partial \phi_1}{\partial \mathbf{n}} = \frac{\partial \phi_1}{\partial x} \frac{\partial x}{\partial \mathbf{n}} + \frac{\partial \phi_1}{\partial y} \frac{\partial y}{\partial \mathbf{n}}. \quad (13)$$

If  $|AB|$  is the length AB,

$$\frac{\partial x}{\partial \mathbf{n}} = \frac{(y_A - y_B)}{|AB|}$$

and

$$\frac{\partial y}{\partial \mathbf{n}} = -\frac{(x_A - x_B)}{|AB|}$$

Substituting into Eq. (13),

$$\frac{\partial \phi_1}{\partial \mathbf{n}} = \frac{\partial \phi_1}{\partial x} \frac{(y_A - y_B)}{|AB|} - \frac{\partial \phi_1}{\partial y} \frac{(x_A - x_B)}{|AB|}. \quad (14)$$

Now, using Eq. (7) and equating node 1 of Fig. 2 to node C of Fig. 3,

$$\begin{aligned} \frac{\partial \phi_1}{\partial x} &= \frac{(b_C \phi_C + b_A \phi_A + b_B \phi_B)}{2\Delta_1} \\ &= \frac{1}{2\Delta_1} [(y_A - y_B)\phi_C + (y_B - y_C)\phi_A + (y_C - y_A)\phi_B] \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \phi_1}{\partial y} &= \frac{(c_C \phi_C + c_A \phi_A + c_B \phi_B)}{2\Delta_1} \\ &= \frac{1}{2\Delta_1} [(x_B - x_A)\phi_C + (x_C - x_B)\phi_A + (x_A - x_C)\phi_B]. \end{aligned}$$

Substituting into Eq. (11) with Eq. (13),

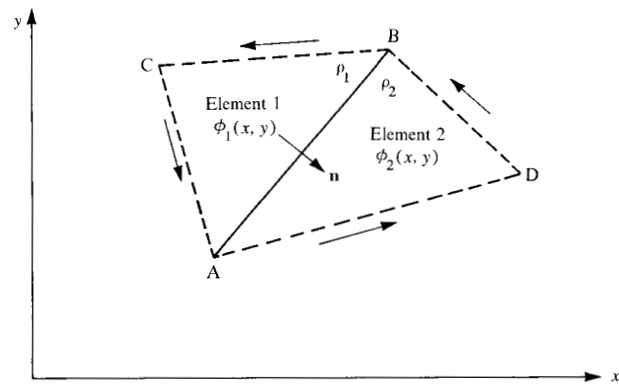


Figure 3 Calculation of total current crossing side AB between two adjacent triangular elements.

$$\begin{aligned} I_{12} = \frac{1}{2\rho_1 \Delta_1} \int_A^B \left\{ \frac{(y_A - y_B)}{|AB|} [(y_A - y_B)\phi_C + (y_B - y_C)\phi_A \right. \\ \left. + (y_C - y_A)\phi_B] - \frac{(x_A - x_B)}{|AB|} [(x_B - x_A)\phi_C \right. \\ \left. + (x_C - x_B)\phi_A + (x_A - x_C)\phi_B] \right\} dl. \end{aligned}$$

Integrating and rearranging items,

$$\begin{aligned} 2\rho_1 \Delta_1 I_{12} &= (y_A - y_B)[(y_A - y_B)\phi_C + (y_B - y_C)\phi_A \\ &\quad + (y_C - y_A)\phi_B] - (x_A - x_B)[(x_B - x_A)\phi_C \\ &\quad + (x_C - x_B)\phi_A + (x_A - x_C)\phi_B]. \end{aligned} \quad (15)$$

Similarly, considering element 2 and Eq. (12),

$$\begin{aligned} 2\rho_2 \Delta_2 I_{12} &= (y_A - y_B)[(y_B - y_A)\phi_D + (y_A - y_D)\phi_B \\ &\quad + (y_D - y_B)\phi_A] - (x_A - x_B)[(x_A - x_B)\phi_D \\ &\quad + (x_D - x_A)\phi_B + (x_B - x_D)\phi_A]. \end{aligned} \quad (16)$$

Adding Eqs. (15) and (16) and eliminating common terms,

$$\begin{aligned} 2(\rho_1 \Delta_1 + \rho_2 \Delta_2) I_{12} &= (y_A - y_B)[(y_A - y_B)(\phi_C - \phi_D) \\ &\quad + (y_D - y_C)(\phi_A - \phi_B)] \\ &\quad - (x_A - x_B)[(x_A - x_B)(\phi_D - \phi_C) \\ &\quad + (x_C - x_D)(\phi_A - \phi_B)], \end{aligned}$$

or

$$\begin{aligned} I_{12} &= \frac{1}{2(\rho_1 \Delta_1 + \rho_2 \Delta_2)} \{ |AB|^2 (\phi_C - \phi_D) \\ &\quad - [(y_A - y_B)(y_C - y_D) \\ &\quad + (x_A - x_B)(x_C - x_D)] (\phi_A - \phi_B) \}. \end{aligned} \quad (17)$$

Equation (17) gives the total current crossing the boundary of two triangular elements in terms of their

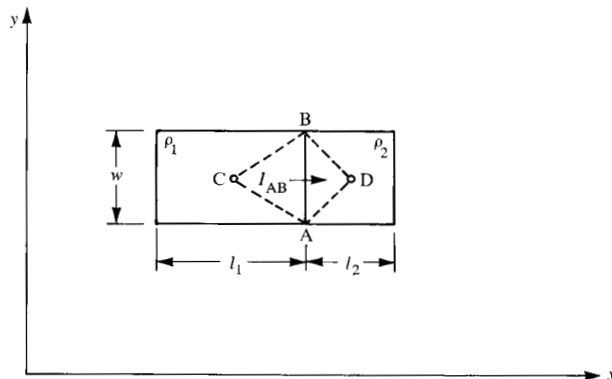


Figure 4 Two adjacent rectangular elements. Equation (22) gives the total current crossing their common side.

square resistances, areas, x-y coordinates, and functional values (voltages) of all their vertices.

For any closed curve  $C$  within a solution domain, the following line integral is constant:

$$\int_C \frac{\partial \phi}{\partial n} dC = \text{constant}. \quad (18)$$

The constant represents the total current generated within the closed curve. Applied to a triangular element (Fig. 2) and using Eq. (17), the integral can now be calculated as

$$\int_C \frac{\partial \phi}{\partial n} dC = \sum_{i=1}^3 I_i = \text{constant}, \quad (19)$$

where 3 represents the sides of the triangle.

Thus the final triangular element equation can be written as

$$I_{\text{element}} = \mathbf{K}^T \boldsymbol{\phi} = \text{constant}, \quad (20)$$

where  $\mathbf{K}$  is a vector of precalculated constants and  $\boldsymbol{\phi}$  a vector of voltage values at the vertices. In this case the length of each vector is six.

#### Rectangular elements

The general equation expressing the total current crossing a random line AB, Eq. (17), will be used here to derive the rectangular element equation. Consider two rectangular elements 1 and 2 with geometrical centers at C and D as shown in Fig. 4. It is easy to see that the total current crossing their common side AB is

$$I_{AB} = \frac{1}{2(\rho_1 \Delta_1 + \rho_2 \Delta_2)} |AB|^2 (\phi_C - \phi_D), \quad (21)$$

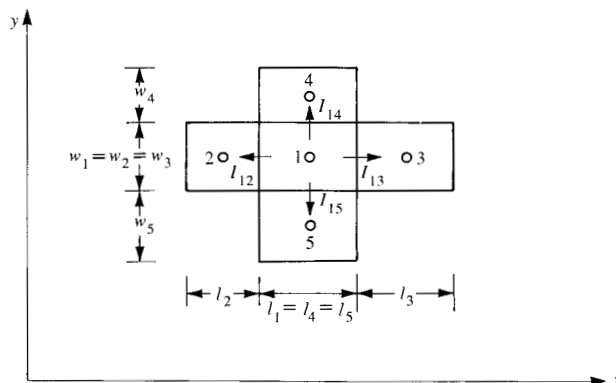


Figure 5 Calculation of net current [Eq. (23)] for a rectangular element bounded by rectangular elements.

or, after substituting for the areas and simplifying,

$$I_{AB} = \frac{2w}{(\rho_1 l_1 + \rho_2 l_2)} (\phi_C - \phi_D). \quad (22)$$

As in the triangular element, the net current around the closed curve of a rectangular element must be constant [Eq. (18)]. Thus, applying Eq. (22) to the four sides of the element in Fig. 5 and summing up,

$$\begin{aligned} I_{\text{element 1}} = & \frac{2w_1}{(\rho_1 l_1 + \rho_2 l_2)} (\phi_1 - \phi_2) \\ & + \frac{2w_1}{(\rho_1 l_1 + \rho_3 l_3)} (\phi_1 - \phi_3) \\ & + \frac{2l_1}{(\rho_1 w_1 + \rho_4 w_4)} (\phi_1 - \phi_4) \\ & + \frac{2l_1}{(\rho_1 w_1 + \rho_5 w_5)} (\phi_1 - \phi_5) = \text{constant}, \end{aligned} \quad (23)$$

or, in general,

$$I_{\text{element}} = \mathbf{K}^T \boldsymbol{\phi} = \text{constant}, \quad (24)$$

where the lengths of the vectors are now five, provided the element is bounded on all four sides by rectangular elements. If a rectangular element is adjacent on any of its sides to a triangular element, then the current crossing that side will be calculated using Eq. (17), while Eq. (22) will be used for the current crossing the other sides (adjacent to rectangular elements). The final element equation in this case will connect seven nodal values, that is, the lengths of  $\mathbf{K}$  and  $\boldsymbol{\phi}$  will be seven.

#### Writing the nodal equations

To form the nodal equation for a rectangular nodal point at the center point of the rectangle, the rectangular element equation is used. For a triangular nodal point lo-

cated at a vertex of the triangle, the nodal equation is formed by considering all the triangular elements that have the nodal point as a common vertex.

An important task of the "automatic element generator" is to create enough triangular elements so that a set of  $n$  linearly independent nodal equations can be formed, where  $n$  is the number of nodal points.

### Solving the system equations

The schemes for solving the resultant linear equations can be generally classified as direct or iterative. The two main considerations in choosing a method are the storage requirements and the computer time. The most commonly used iterative scheme is the Gauss-Seidel or successive overrelaxation (SOR) method in which the solution vector is obtained after a number of iterations employing the general relation

$$[\phi]_K = [\phi]_{K-1} + \omega[\Delta\phi]_{K-1,K}, \quad (25)$$

where  $K$  is the iteration number and  $\omega$  a relaxation factor used to speed up convergence. In the case of finite difference or finite element methods,  $\omega$  depends on the particular geometry involved. The storage requirement is directly proportional to the number of nodal points and the method uses much less storage than the direct approach. Unfortunately, the same is not true with the calculation time required to obtain a solution. Unlike the direct method, the computer time depends on the geometry under consideration and on the boundary conditions defining the particular problem. Integral constant-current boundary conditions slow down convergence to the point where the computer time required is hard to predict. In contrast, the direct approach, which is more expensive in storage requirements, is much more efficient in terms of computer time. It is related to the number of nodal points and the bandwidth of the matrix and independent of the boundary conditions. One of the main reasons the direct method was chosen is that once one solution is obtained for a given geometry, multiple solutions for other boundary conditions may be obtained with very little additional computer time. The problem can be written as

$$\mathbf{Ax} = \mathbf{b}, \quad (26)$$

where

$\mathbf{A} = [A_{ij}] = n \times n$  matrix of coefficients,

$\mathbf{x}$  = nodal solution vector, and

$\mathbf{b}$  = nodal boundary condition vector.

Matrix  $\mathbf{A}$  is sparse, symmetric, and positive definite, and thus the LU decomposition can be written as

$$\mathbf{A} = \mathbf{U}^T \mathbf{D} \mathbf{U}. \quad (27)$$

The following triangular systems, representing the forward reduction of  $\mathbf{b}$  and back substitution,

$$\mathbf{U}^T \mathbf{D} \mathbf{y} = \mathbf{b}, \quad (28)$$

$$\mathbf{U} \mathbf{x} = \mathbf{y}, \quad (29)$$

are next solved for the solution vector  $\mathbf{x}$ .

The finite element equations are usually very sparse, typically less than 0.1 percent non-zero, while the systems to be solved are, in most applications, extremely large, where 10 000 or more nodes are not unusual. References [6-8] discuss in general large sparse systems of equations. To minimize fill-in (generating of non-zero terms) during the decomposition of  $\mathbf{A}$  in Eq. (27), the ordering of the equations into an almost banded form seems to be the most logical and efficient approach.

Some of its advantages are the following: 1) the ability to easily predict fill-in and, therefore, use a static storage scheme, 2) the minimum overhead storage required for decomposition, and 3) the lack of any decision making during factorization (no pivoting or checking for zeros). The most important advantage, however, is that the method is well adapted to using an external storage device, which is imperative for large systems. Thus, minimum main memory is required.

The separate parts of the sparse matrix analysis are given next.

### • Optimal node numbering

The objective of this part is to renumber the nodal points so that the resulting assembly matrix  $\mathbf{A}$  is banded and has a minimum bandwidth. A novel node numbering scheme which accomplishes this optimally is described in the following steps.

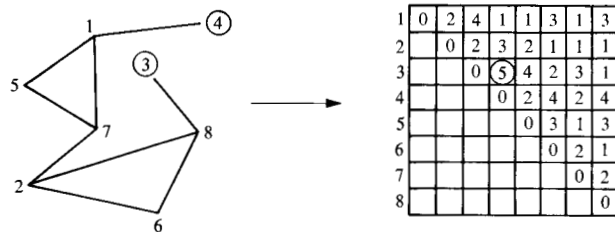
Construct an undirected graph of the nodal points, connecting any two nodal points with an edge only when the nodal points in question appear in the same nodal equation. Form the incident matrix  $\mathbf{c}$  with  $c_{ii} = 0$ . Form a distance matrix  $\mathbf{d}$  and enter only the one-edge connections, that is, initially  $d_{ij} = c_{ij}$ . Calculate  $\mathbf{c}^2, \mathbf{c}^3, \dots, \mathbf{c}^k$  by performing the following three steps repeatedly until the distance matrix is filled up:

$$\mathbf{c}_{ij}^k = \bigvee_{m=1}^n (c_{im}^{k-1} \wedge c_{mj}) \quad (n = \text{matrix size}).$$

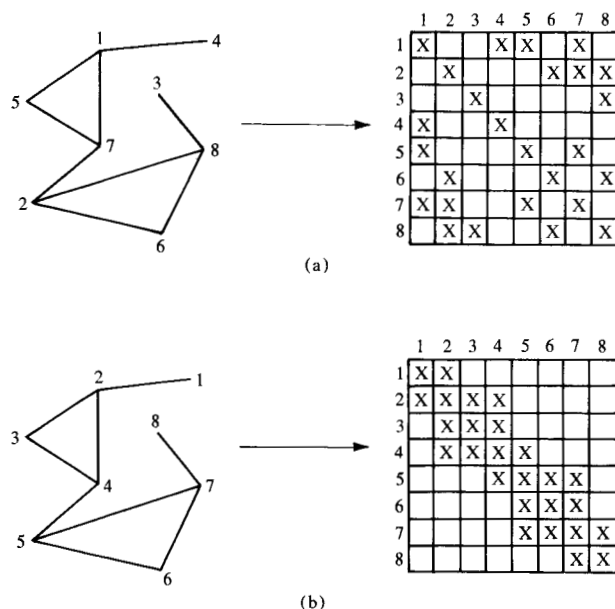
If  $d_{ij} = 0$  and  $c_{ij}^k \neq 0$  set  $d_{ij} = k$ .

$$\text{Set } c_{ij}^k = 0 \quad \text{for all } d_{ij} = k - 1. \quad (30)$$

The minimum distances (distance not including loops) of all pairs of nodal points have now been obtained. Starting



**Figure 6** Resulting distance matrix for a small sample problem. Points 3 and 4, which are the farthest apart (distance 5), are the best starting points for renumbering. Initial numbering is 4 1 5 7 2 6 8 3. Optimal numbering is 1 2 3 4 5 6 7 8.



**Figure 7** Example showing reduction of bandwidth from seven to four after renumbering: (a) Matrix  $A$  before renumbering. (b) Matrix  $A$  after renumbering.

with any one of the most distant points (preferably one with fewer degrees of freedom) we proceed by renumbering the points in succession following their distance pattern. That is, the next point to be numbered will always be the one closest to the previously numbered point. Figure 6 gives the distance matrix for a small sample problem with the optimal renumbering of the points while Fig. 7 shows the improvement in the bandwidth of matrix  $A$  for the same problem. The reader can recognize that the method locates the best starting point to renumber the nodes. This point is not, as is sometimes stated [9], a point of minimum degree of freedom.

Another less expensive heuristic procedure which in most cases will produce the minimum bandwidth is as follows. Form the graph as above, and, starting at a random node, number the nodes following the connectivity of the graph and only proceed to a new node after all the one-edge connections to the previously numbered node have been numbered. When all the nodes are numbered, repeat the process, starting this time with the last numbered node. The first and last numbered nodes this time are often the nodes with the maximum distance (see Fig. 8) and therefore represent the best starting points. This method will not produce the minimum bandwidth for some random starting points in some symmetric graphs.

#### ● Calculation of variable semi-bandwidth of $A \rightarrow U$ and its total storage requirement

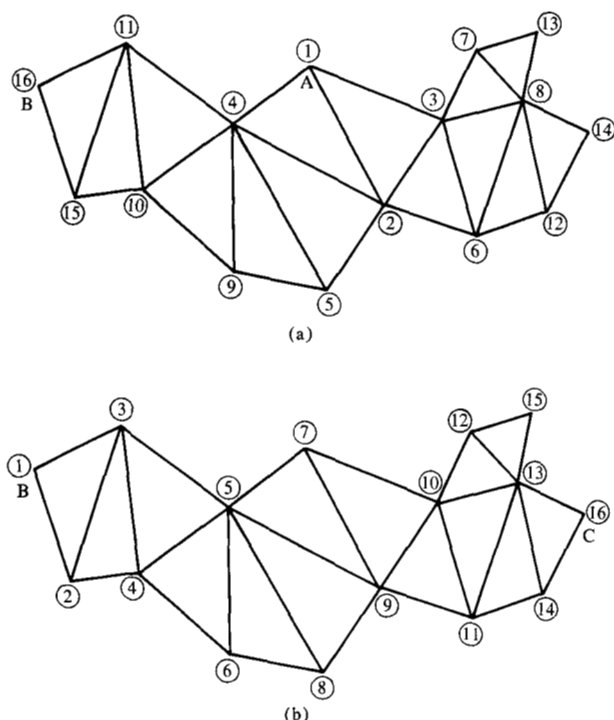
Let  $2w + 1$  be the bandwidth of  $A$ . Since  $A$  is symmetric, we need deal only with the semi-band  $w + 1$  (including the diagonal). All non-zero terms of  $A$ , as well as the generated non-zero terms of the upper triangular factor  $U$ , will be included within a banded matrix of maximum size  $n(w + 1)$ . During decomposition (Eq. 27), the above banded matrix  $A$  is transformed into the factor  $U$  so that the same storage space is used. After decomposition,  $A$  is not needed. A modification of the standard banded approach was made here which resulted in a considerable reduction of both computation time for factorization and storage requirement for  $U$ .

Instead of a constant semi-bandwidth  $w + 1$  (distance of most remote non-zero entry of  $A$  from the diagonal), a variable width ( $w_i, i = 1$  to  $n$ ) is calculated during decomposition, considering the location of the non-zero terms in  $A$  and the location of the most remote generated non-zeros in  $U$ . These locations are known before decomposition starts, so that a static storage scheme can be used. Figure 9 shows the original non-zeros in  $A$  ( $\otimes$ ) and the generated non-zeros in  $U$  ( $\times$ ). The dotted line is the profile of the final factor  $U$ . The total storage required for  $U$  is precalculated as the summation of the lengths of the equations

$$\sum_{i=1}^n w_i \quad (31)$$

#### ● Partitioning of banded matrix $A \rightarrow U$ into records

In this part of the program, a decision to use external storage or not is made depending on the size of  $U$ . If enough main region size is provided for  $U$  (dimension of  $U \geq \sum_{i=1}^n w_i$ ), this part is bypassed. Otherwise the banded matrix is partitioned into a number of records and placed in an external storage device. The record size and therefore the total number of records depends on the size of the  $U$  array provided by the user. This size (the dimension of  $U$ ) must be at least as large as  $n$ .



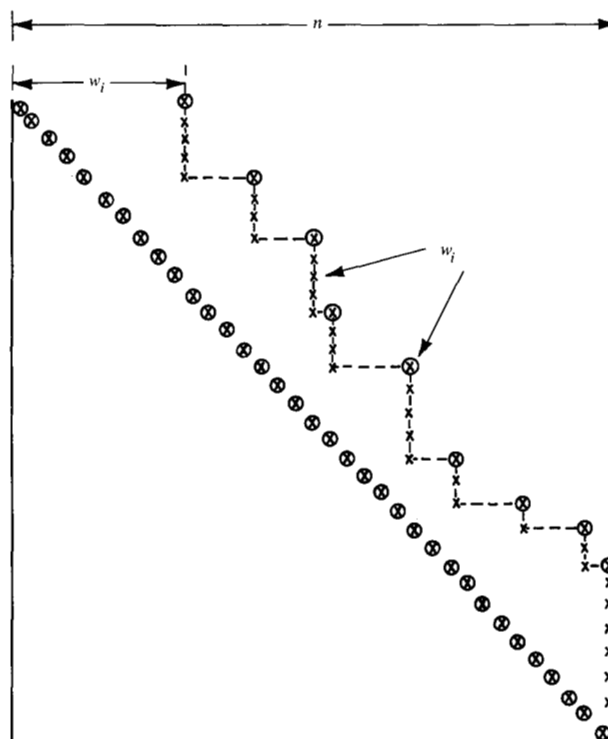
**Figure 8** Example illustrating the heuristic method for producing minimum bandwidth. Points B and C are the best starting points for renumbering: (a) First iteration. Starting point A is a random point. Point B is the last numbered point and will be the starting point for iteration two. (b) Second iteration. Starting point for the numbering scheme is point B.

It is fortunate, as can be seen from the Appendix, that not all of  $A$  is required in main storage during the decomposition phase. In Fig. 10, only the section ABC is required in main in order to process the first record AB (shaded area). When the first record is processed, records 2 and 3 are shifted upwards in main and record 4 is brought in. Record 1 is placed back on the disk. Next record 2 is decomposed. This process continues until all records are decomposed and placed on the disk. The solution of Eqs. (27), (28), and (29) appears in the Appendix.

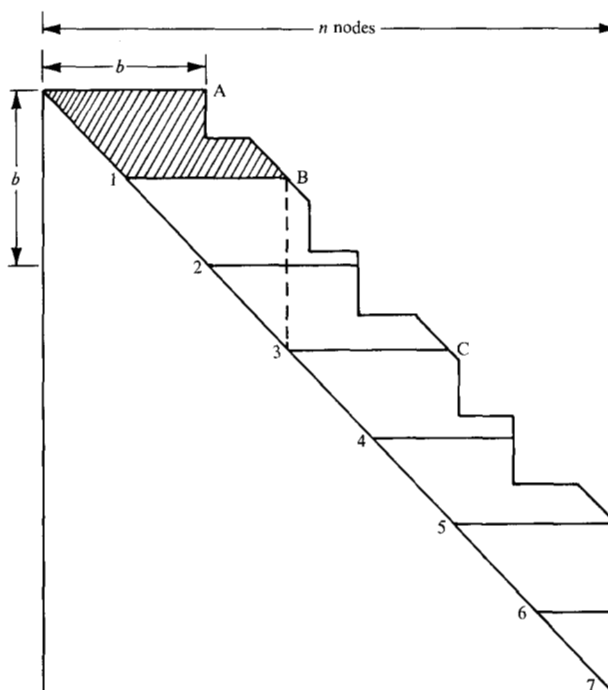
A method has been described which will produce the potential distribution for any given irregular plane geometry and boundary conditions (voltages and/or currents). The next section will describe how this part is used to obtain the equivalent resistance network for any power plane with any number of terminals.

#### Calculation of equivalent resistance network

Before stating the objective of this part of the package, let me define as ports of a given plane geometry those locations in the plane through which current is entering or



**Figure 9** Decomposition of banded matrix  $A$  showing the generation of non-zeros (fill-in) in  $U$ :  $\otimes$ , original non-zeros in  $A$ ;  $\times$ , fill-in.



**Figure 10** Symmetric banded matrix  $A$  divided into records. Bandwidth =  $2b + 1$ . Shaded area AB represents one record. Three records (down to C) are required in main storage in order to process the first record.



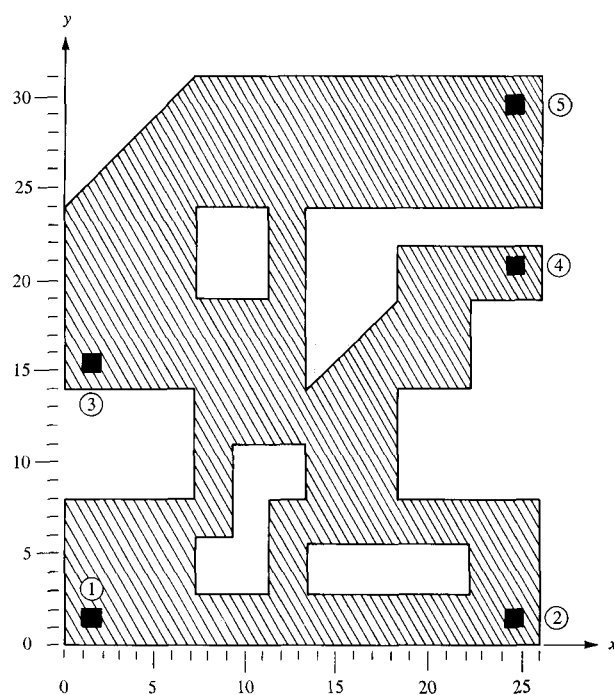


Figure 11 Diagrammatic example of a contacting plane with five ports.

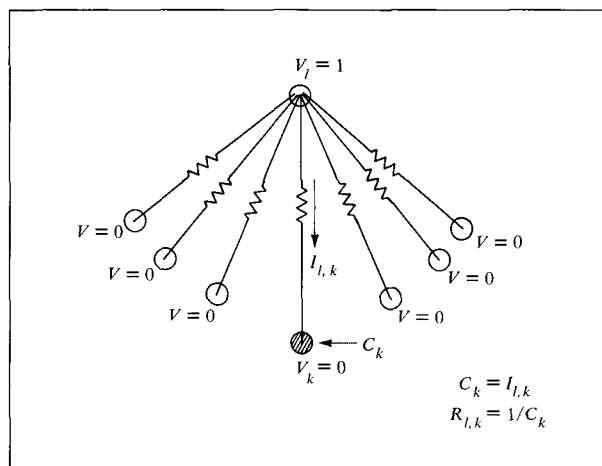


Figure 12 Setting boundary conditions. The total current  $C_k$  leaving port  $k$  is calculated by evaluating the line integral around the port once the field solution has been obtained.

leaving the plane. The reader should not confuse the ports with the elements. A port is in itself a plane geometry and may consist of one or many adjacent elements. It represents an area of contact with the plane. Figure 11 gives an example of a plane with five ports. For a complete and accurate characterization of the given current contacting plane (equivalent resistance network), the resistances of

all branches connecting all pairs of ports are required. That is, for  $P$  ports, the complete resistance network will consist of  $P(P-1)/2$  resistances. It is true that, depending on the geometry under consideration, some of these resistances will be very large, indicating that there is no current flowing in that branch, which can therefore be ignored. Thus the objective of this part is to calculate the equivalent resistance network. To accomplish this,  $P-1$  independent potential solutions are required. Since the resistances of the equivalent network are independent of the kind of boundary conditions imposed, the program sets the boundary conditions (voltages at ports) in such a way that, when a potential solution is obtained, a number of resistances can be calculated directly.

Let  $c_i$  be the total current entering the plane through port  $i$  and  $I_{ij}$  the branch current connecting port  $i$  to port  $j$ . Then, for a complete resistance network of  $P$  ports,

$$c_i = \sum_{\substack{j=1 \\ j \neq i}}^P I_{ij} \quad (32)$$

By choosing proper boundary conditions, a solution can be obtained where, for any port  $k$ , all branch currents  $I_{k,j}$  ( $j = 1, P, j \neq k$ ) are zero except one,  $I_{l,k}$  (Fig. 12).

Consider, for instance,

Voltage at port  $l$ ,  $V_l = 1$ .

Voltages at all other ports,  $V_i = 0 \quad i = 1, P; i \neq l$ .

Then the solution obtained will satisfy

$$c_i = I_{l,i}, \quad i = 1, P; i \neq l,$$

and therefore

$$R_{l,i} = \frac{\Delta V_{l,i}}{c_i} = \frac{1}{c_i} \quad i = 1, P; i \neq l. \quad (33)$$

The total current  $c_i$  through the port  $i$  is calculated by evaluating the line integral

$$\int_c \frac{\partial \phi}{\partial \mathbf{n}} d\mathbf{c},$$

where  $c$  is the boundary of the port. The finite element equations derived in the second section are used.

Thus, the calculation of all resistances for  $P$  ports is accomplished by executing repeatedly the following sequence:

- For port  $l$ , set  $V_l = 1$ ,  $V_1$  to  $V_P = 0$ .
- Obtain potential solution and calculate  $c_i$ ,  $i = 1$  to  $P$ .
- Calculate using Eq. (33)  $R_{l,i}$ ,  $i = 1, P; i \neq l$ .

Repeat with  $l$  from 1 to  $(P-1)$ .

An important advantage of the direct method is its efficiency in producing multiple solutions for the same geometry and different boundary conditions. Since the sparse matrix  $\mathbf{A}$  depends only on the geometries involved, its factorization [Eq. (27)], which represents most of the work required to obtain a solution, is done only once. That is,  $\mathbf{U}$  remains unchanged and any new solution is obtained by merely updating the vector  $\mathbf{b}$  and executing Eqs. (28) and (29).

The last section deals with the application of this part in solving a multi-plane configuration.

### DC analysis of a multi-plane configuration

A number of electrically insulated planes, connected to one another through vias of known resistances, can represent many present-day packaging designs in LSI technology. To obtain the potential distribution and accurately evaluate any of the given plane geometries, the boundary conditions, that is, voltages or currents of all its contact points (ports), must be known. These conditions are in general unknown. Consider for instance the multi-plane structure illustrated in Fig. 13. The only known conditions are the incoming voltage at point 1 and the outgoing currents at points 18, 19, 20, and 21. No detailed analysis of plane 2 or any other plane can be performed without knowing how much current is carried by each of the connecting rods. In the case of plane 2, for example, the conditions at all its contact points, 8 through 13, must be known.

A step-by-step procedure is presented here which resolves the problem and permits the accurate evaluation of any of the given plane geometries.

- Step 1. Obtain the equivalent resistance networks (see previous section) for all given plane geometries. The reader is reminded that no boundary conditions are required for these calculations. The only information needed is the geometries involved and their square resistances.
- Step 2. Merge the above resistance networks with the via resistances and obtain an equivalent resistance network representing the multi-plane configuration (Fig. 14).
- Step 3. Using dc resistance network equations, solve the multi-plane resistance network obtained above for the given boundary conditions [10]. In the case of the example of Figs. 13 and 14, these boundary conditions are  $V_1$ ,  $I_{18}$ ,  $I_{19}$ ,  $I_{20}$  and  $I_{21}$ . The reader must notice that the currents are not currents flowing on the branches of the network but the net currents of the contact points. Thus the conditions at the remaining points are  $I_{\text{net}} =$

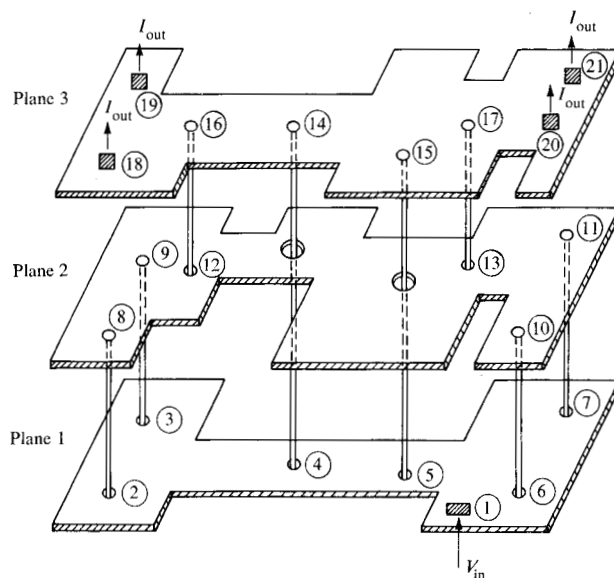


Figure 13 Multi-plane configurations illustrating a general packaging design.

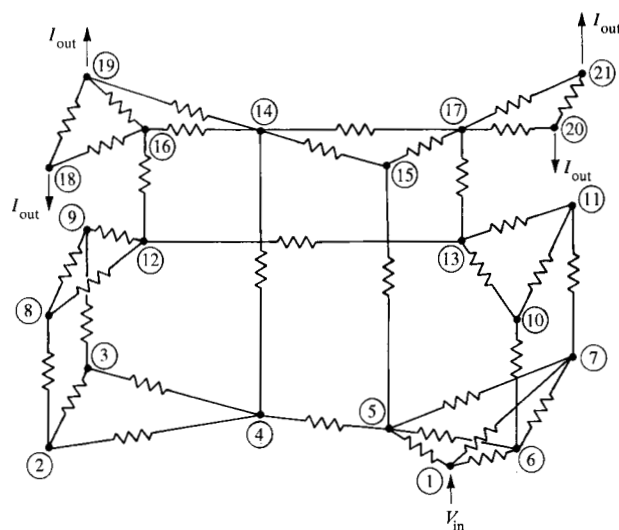


Figure 14 Equivalent resistance network for multi-plane configuration of Fig. 13.

0. Application of the network equations results in a sparse symmetric system and the technique of the section on solving the system of equations is used to obtain its solution. This time the unknowns are either the voltages or the net currents at the contacts.

- Step 4. The boundary value problem for any of the planes involved is now well defined since the bound-

ary conditions (voltages at contacts) are all known. Its solution will produce the field distribution for the particular plane.

The complete software package can include some subsequent calculations like current densities and maximum voltage drops per plane, amount of current per via, etc.

### Summary

The trend to ever higher levels of integration and the need to utilize all available space during any level of packaging usually results in very complex geometrical shapes. Geometries of power buses or redistribution planes with hundreds of vertices are not uncommon. Speed of calculations and storage requirements remain the two main considerations in developing any packaging analysis tool, and the method presented here tries to meet both requirements without sacrificing accuracy of calculation. In formulating the finite element solution, triangular elements, being more expensive in computation time, are generated only where the geometries under consideration necessitate them. By considering the  $x$ - $y$  orientation of the rectangular elements, it is possible to simplify their equations and derive nodal equations relating only the geometrical centers of the neighboring rectangular elements. The choice of the direct method in solving the system of equations was shown to be very appropriate, especially when multiple solutions are required. The bandedness of the sparse matrix is assured through the use of the optimal numbering scheme and is solved with a minimum expenditure of bookkeeping and decision-making time. Most importantly, the solution technique is independent of the main storage available. Finally, the special nature of the designs involved in packaging was recognized; an approach was introduced which reduces a three-dimensional problem to a problem requiring a number of two-dimensional solutions.

### Acknowledgments

The author wishes to express his appreciation to W. C. Bakker for his help in the preparation of this paper and numerous discussions on the subject, to B. C. Hempel for his programming assistance in coding and developing the optimal numbering algorithm, and to J. E. Hespenheide for his work in linking this tool to a general engineering design system. The author also wishes to express his gratitude to his management for their support of this project.

### Appendix: Sparse matrix solution

Solutions of large sparse systems are discussed in, *e.g.*, Refs. [5, 11, 12]. Presented in this Appendix is the detailed particular sparse matrix solution for the modified banded approach developed in this paper.

### • Matrix decomposition

The Gauss-Dolittle scheme is used to perform the LDU decomposition of  $A$  [Eq. (27)]. This is accomplished algorithmically as follows. Let  $n$  be the number of equations and  $w_i$ ,  $i = 1$  to  $n$ , be the variable semi-bandwidths (including the diagonal). Assuming the band matrix to be arranged as a matrix  $A_{i,j}$ ,  $i = 1$  to  $n$  and  $j = 1$  to  $w_i$ , then perform

$$A_{k+i,j} = A_{k+i,j} - \left[ \frac{A_{k,i+1}}{A_{k,1}} \right] A_{k,i+j}, \quad (A1)$$

$$A_{k,i+1} = \frac{A_{k,i+1}}{A_{k,1}}, \quad (A2)$$

for

$$k = 1 \text{ to } n - 1,$$

$$i = 1 \text{ to } w_k - 1,$$

$$j = 1 \text{ to } w_k - i.$$

As mentioned above, the location of the farthest non-zero entry of the band matrix (either original or created during decomposition) determines the length ( $w_i$ ) of the  $i$ th equation within the band. This length is precalculated, and

$$w_i \leq w_{\max} \quad i = 1 \text{ to } n,$$

where  $w_{\max}$  is the maximum element of the vector  $w$ . The storage saved due to this modification is

$$w_{\max} \times n - \sum_{i=1}^n w_i. \quad (A3)$$

Considerable calculation time is also saved due to the creation of a profile semi-bandwidth instead of the maximum  $w_{\max}$ . At the end of this step,  $A$  has been transformed to the upper triangular factor  $U$  (same storage space for  $A$  and  $U$ ).

### • Forward reduction [Eq. (28)]

Once the triangular factor  $U$  has been obtained, the forward reduction of  $b$  is accomplished by performing

$$b_{k+i} = b_{k+i} - U_{k,i+1} \times b_k, \quad (A4)$$

$$b_k = \frac{b_k}{U_{k,1}}, \quad (A5)$$

for

$$k = 1 \text{ to } n - 1,$$

$$i = 1 \text{ to } w_k - 1,$$

and

$$b_n = \frac{b_n}{U_{n,1}}, \quad \text{for } k = n. \quad (A6)$$

• *Back substitution [Eq. (29)]*

The final step in obtaining the solution is

$$b_k = b_k - U_{k,i+1} \times b_{k+i},$$

for

$$k = n - 1 \text{ to } 1,$$

$$i = 1 \text{ to } w_i - 1.$$

The given right-hand-side vector **b** has been transformed and now represents the solution vector **x** (same storage space used for **b** and **x**).

## References

1. G. D. Smith, *Numerical Solution of Partial Differential Equations*, Oxford University Press, London, 1965.
2. P. Tong and J. N. Rossettos, *Finite-Element Method*, MIT Press, Cambridge, MA, 1967.
3. O. C. Zienkiewicz, *The Finite Element Method in Structural and Continuum Mechanics*, McGraw-Hill Publishing Co., Ltd., London, 1967.
4. K. H. Huebner, *The Finite Element Method for Engineers*, John Wiley & Sons, Inc., New York, 1975.
5. W. F. Tinney and J. W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," *Proc. IEEE* (Special Issue on Computer-Aided Design) **58**, 1801-1809 (1967).
6. F. G. Gustavson, W. Liniger, and R. Willoughby, "Symbolic Generation of an Optimal Crout Algorithm for Sparse Systems of Linear Equations," *J. ACM* **17**, 87-109 (1970).
7. Hsueh Y. Hsieh, "Pivoting-Order Computation Method in Large Random Sparse Systems," *IEEE Trans. Circuits Systems CAS-21*, 225 (1974).
8. Hsueh Y. Hsieh, "Fill-in Comparisons Between Gauss-Jordan and Gaussian Eliminations," *IEEE Trans. Circuits Systems CAS-21*, 230 (1974).
9. E. Cuthill and J. McKee, "Reducing the Bandwidth of Sparse Symmetric Matrices," *ACM Proceedings of 24th National Conference*, New York, 1969.
10. H. E. Brown, *Solution of Large Networks by Matrix Methods*, John Wiley & Sons, Inc., New York, 1975.
11. A. Jennings, *Matrix Computations for Engineers and Scientists*, John Wiley & Sons, Ltd., London, 1977.
12. J. R. Bunch and D. J. Rose, Eds., *Sparse Matrix Computations*, Academic Press, Inc., New York, 1976.

Received April 5, 1979; revised July 2, 1979

The author is located at the IBM Data Systems Division laboratory, East Fishkill (Hopewell Junction), New York 12533.